GIS ANCHORED INTEGRATED PLANTATION MANAGEMENT: TEA

(A Package Developed Under Research Promotional Scheme of All India Council for Technical Education)

Developed by —

Puspendu Sarkar

Arpita Nath

Atanu Moitra

Dept. of Computer Science & Application

University of North Bengal

Under Principal Investigation of —

Dr. J. K. Mandal,

Professor & Head,

Department of Computer Application

Kalyani Government Engineering College

Kalyani, Nadia, West Bengal

Acknowledgement

All works of the project had been carried out by the students of Master of Computer Application, 3rd year (final semester), North Bengal University for the partial fulfillment of the Master of Computer Application Degree.

I acknowledge my heartful of gratitude to Dr. Jyotsna Kumar Mandal, Principal Investigator of the Project and also Professor & Head, Department of Computer Application, Kalyani Government Engineering College, Kalyani, Nadia, under whose supervision and proper guidance the entire project work had been undergone.

Many other individuals have contributed in the development of this project. With so many people to be thanked for their valuable comments and exclusive suggestions, I would like to express my deep gratitude to them for providing us with the computer laboratory for accessing the internet, and testing the project proceedings when required. I would also like to thank the management of Phuguri Tea Estate who have helped providing all the necessary information regarding the tea estate.

My sincere gratitude to all the faculties, Non-Teaching staffs and friends for sharing their knowledge, advice and expertise during the development of the project and devoting their valuable time to make this project a success.

PUSPENDU SARKAR
ARPITA NATH
ATANU MOITRA

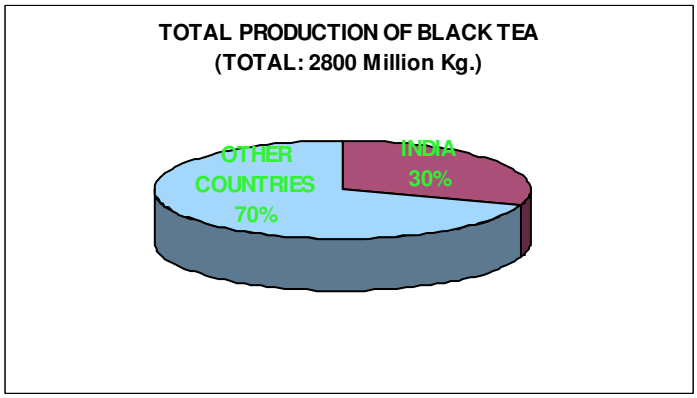
Contents

1

Introduction

Introduction:

Tea is one of the most important and highest foreign exchange earning agricultural product of India. The Indian states of Assam, Meghalaya, Tripura, North Bengal (Darjeeling) and Sikkim contribute significantly to the overall tea production in the country. Tea is also grown in Southern part of the country such as TamilNadu, Karnataka and Kerala. Total Tea cultivation area is approximately 0.45 million hector. There are three major varieties of tea produced world wide such as Orthodox, CTC, & Green Tea. India mainly produces Black Tea whereas Green Tea is very popular in Japan, Taiwan, and China etc. India produces 850 millions Kgs. of Tea annually, which is 30 percent of world production (Annual world production 2800 million Kgs.). For last couple of years, due to rapid changes in global economic scenario, globalization of Indian economy, breaking of USSR and increasing presence of countries like Kenya, Indonesia, Sri Lanka etc. in global export market, Indian tea industry is at the cross roads. Consequently average price realized per Kg. of tea by Indian industries decreasing steadily as described in the below figure.



Industry and Government has realized that there is immediate need to modernize the tea production process to make it globally competitive. The major national objective of tea industry is to utilize available lands on estates towards sustainable development by ensuring land use efficiency with proper soil conservation measures and environment protection. In addition, producers' objective is to manage tea estates to maximize profits. Therefore, the overall objective of this project is to identify suitable strategic land use development plan and to introduce precision farming to maximize profitability. The strength of **IT** in the form of **GIS** and **ERP** solution is expected to give excellent results towards achieving this goal.

2

Overview of the Project

2.1 Project Details

Project Title	**: GIS Anchored Integrated Plantation Management: Tea (*A Package Developed Under Research Promotional Scheme of All India Council of Technical Education*)**
Project Guide	**: Dr. J. K. Mandal, Principal Investigator. (Professor and Head of the Dept. of C.A., KGEC).**
Place of Work	**: Kalyani Government Engineering College, Kalyani, Nadia, West Bengal.**
Work Field	**: Phuguri Tea Estate, Mirik, Darjeeling, W.B.**
Project Duration	**: Five Months (5th January, 2006 – 24th May, 2006)**
Platform	**: Windows XP (Professional)**
Software Used	**: J2SDK 1.4.1, MatLab 6.1**
Technology Used	**: Java Swing**
Database	**: Oracle 9i**
Development Tool	**: J-Creator 3.0 LE**
Team Size	**: Three**

2.2 What is GIS?

"Everything is replaced to everything else, but near things related than distant things."

Tobler's First Law of Geography:

Geographic Information System (**GIS**) is a computer based information system used to digitally represent and analyze the geographic features presents on the earth's surface and the events (non spatial attributes linked to the geography under study) that taking place on it. The meaning to represent digitally is to convert analog (smooth line) into a digital form.

A GIS is an information system designed to work with data referenced by spatial/geographic coordinates. In other words, GIS is both a database system with specific capabilities for spatial reference data as well as a set of operation for working with the data. It may also be considered as a higher order map.

GIS technology integrates common database operation such as query and statistical analysis with the unique visualization and geographic analysis benefits make it valuable to a wide range to public and private enterprises for explaining events, predicting outcomes, and planning.

A Geographic Information System is computer-based system, which is used to digitally reproduce and analyze a feature present on earth surface and the events that take place on it. In the light of the fact that almost 70% of the data have geographical reference as its denominators, it becomes imperative to

underline the importance of a system, which can represent the given data geographically.

2.3 Description of the Project:

Up-to-date and reliable information is vital for managerial purposes to take efficient planning decisions. Researchers, policy makers as well as estate managing companies may wish to integrate spatial, temporal, and socio-economic data in order to get better manageability and higher and quality productivity.

GIS is widely accepted as a tool for the establishment of integrating spatial and attribute data. In recent years the development of Geographical Information System (GIS) makes timely accessible of spatial and temporal data. Moreover, its capability of spatial analysis and presentation makes it useful tool for this kind of applications.

Data collection from the field with the help of field sensors and integration of these data to GIS and ERP package will lead to effective and efficient planning for better management.

This GIS Anchored Integrated Plantation Management: Tea package for the Tea Enterprises comprises the followings:

- Detail map of the entire Tea Estate
- Thematic layer maps of the estate
- GIS based front-end
- Attribute ERP database
- Menu-based Spatial query builder

2.4 Objective of the Project:

The Geographical Information System or GIS, in short, is rapidly growing technology for representing spatial data all over the world. Interpretation of database information has always been cumbersome and complex. On the other hand, visual data representation is worth a thousand words.

Our project is mainly focused on the following things:

- Loading the complete image of Tea Garden
- Segmenting that image of the garden into the corresponding sections.
- Create the database related to the Tea Garden and each section.
- Insert, Delete, and Update data of the database.
- Generating the overall report of different garden processes of the garden management
- Retrieve data from the database according to the queries and show the results to the users.

3

Scheme of the Project

3.1 Application Development Environment

3.1.1 Java Database Connectivity (JDBC):

The JDBC is a pure java API (Application Programming Interface) used to execute a SQL statement. It provides a set of classes and interfaces that can be used by developers to write database application. Basic JDBC interaction in its simplest form can be broken down into four steps:

1. Open a connection to the database.
2. Execute SQL statement.
3. Processes the result.
4. Close the connection to the database.

❖ **JDBC Driver Types:**

Sun has defined four JDBC driver types to meets a different application needs.

- ***Type 1:*** **JDBC-ODBC Bridge plus ODBC Driver:**

The first type of JDBC driver is the JDBC-ODBC Bridge. This driver type is provided by Sun with version 1.1 and later. It is a driver that provides JDBC access to database through ODBC driver. The ODBC driver must be configure on the client for the bridge to work. The driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.

- ***Type 2:*** **Native API partly — JAVA Driver:**

The native to API driver converts JDBC commands to DBMS — specific native calls. This is much like the restriction to type 1 driver. The client must have

binary code loaded on its machine. These drivers do have an advantage over type 1 driver because they interface directly with the database.

- ***Type 3:*** **JDBC – Net pure JAVA Driver:**

The JDBC-Net Drivers are three-tire solutions. This type driver translates JDBC calls into a database independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS specific protocol, which is sent to a particular database. This is by far the most flexible JDBC solution.

- ***Type 4:*** **Native-Protocol Pure JAVA Driver:**

The type 4 drivers are pure JAVA drivers that communicate directly with the vendor's database. They do this by converting JDBC commands directly into the database engine's native protocol. The type 4 driver has a very distinct advantage over all the other driver types. It has no additional translation or middleware layer, which improves performance tremendously.

❖ **Getting a connection using JDBC:**

The 1st step in using a JDBC driver to get a database connection involves loading them specific driver class into the application's JVM. This makes the driver available later, when we need it for opening the connection. An easy way to load the driver class is to use the **Class.forName (method).**

Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");

When the driver is loaded into memory, it registers itself with the java.sql.DriverManager class as an available database driver.

The next step is to ask the **DriverManager** class to open a connection to a given database, where the available database is specified by specially formatted URL. The method to open the connection is **DriverManager.getConnection().** It returns a class that implements the java.sql.Connection interface:

Connection connection=DriverManager.getConnection("jdbc: odbc: aaa","project","project");

A JDBC URL identifies an individual database in a driver specific manner. Different Drivers may need different information in the URL to specify the host database.

For example, the **JDBC-ODBC bridge** uses:

jdbc:odbc:datasourcename, odbcoptions.

During the call to **getConnection(),** the DriverManager object asks each register driver if it recognizes the URL. If a driver says yes, the driver manager uses that driver to create the connection object.

❖ **Execution SQL Queries:**

To really use a database, we need to have some way to execute queries. The simplest way to execute query is to use the java.sql.Statement class. Statement objects are never instantiated directly; instead a program calls the **CreateStatement()** method of connection to obtain a new Statement object:

Statements statement = connection.createStatement ();

A query that returns the data can be executed using the **executeQuery()** method of Statement. This method executes the statements and returns a **java.sql.ResultSet** that encapsulates the retrieve data:

ResultSet rs = statement.executeQuery("select sec_area from area where sec_code='pug1';");

We can think of ResultSet object as a representation of the query result returned one row at a time. We use **next()** method of a ResultSet to move from row to row. The ResultSet interface also boasts a multitude of methods that are among the most frequently used for retrieving column values

```
while(rs.next())
{
        String name=rs.getString("sec_area");
}
```

We should know that the ResultSet is linked to its parents statement. Therefore, if a statement is closed or used to execute another query, any related ResultSet object are closed automatically.

❖ **Handling SQL Exception:**

In program catch block throw two exception: ClassNotFoundException and SQLException.

The former is thrown by the Class.forName() method when the JDBC Driver class can not be loaded. The later is thrown by any JDBC method that has a problem. SQLException objects are just like any other Exception type, with the additional feature that they can chain. An example is shown below:

```
Catch(SQLException e)
{
        System.out.println(e.getMessage);
 }
```

❖ **Two-tier Model:**

The JDBC-API supports both **two-tier** models and **three-tier** models for database access. In the two-tier model, a java applet or application talks directly to the data source. This requires a JDBC driver that can communicate with the particular data source being accessed. An application's commands are delivered to the database, and the results of those statements are sent back to the user.

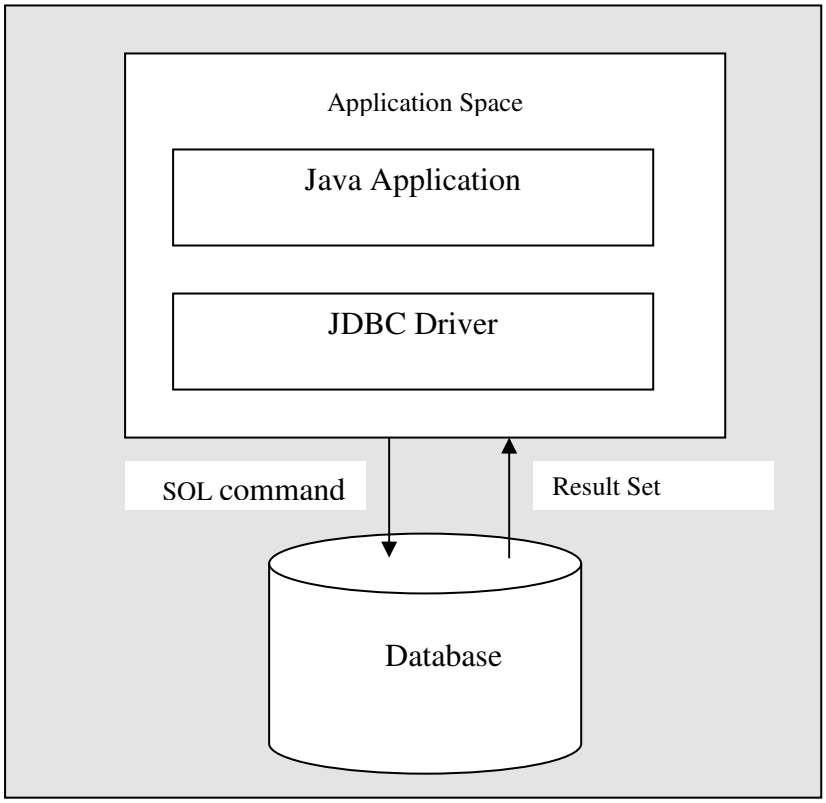


Fig: Two-tier model

❖ **Three- tier Model:**

In the three-tier model, commands are sent to a "middle tier" of services, which then sends the commands to the data source. The middle tier of

three-tier architecture is the application server, which isolates data processing in a central location and maximizes object reuse. It contains problem specific objects called business objects, which look after accounts, customers, and transactions that exist independently of how a user might see them. The data source processes them and sends the results back to the middle tier, which then sends them to the user.

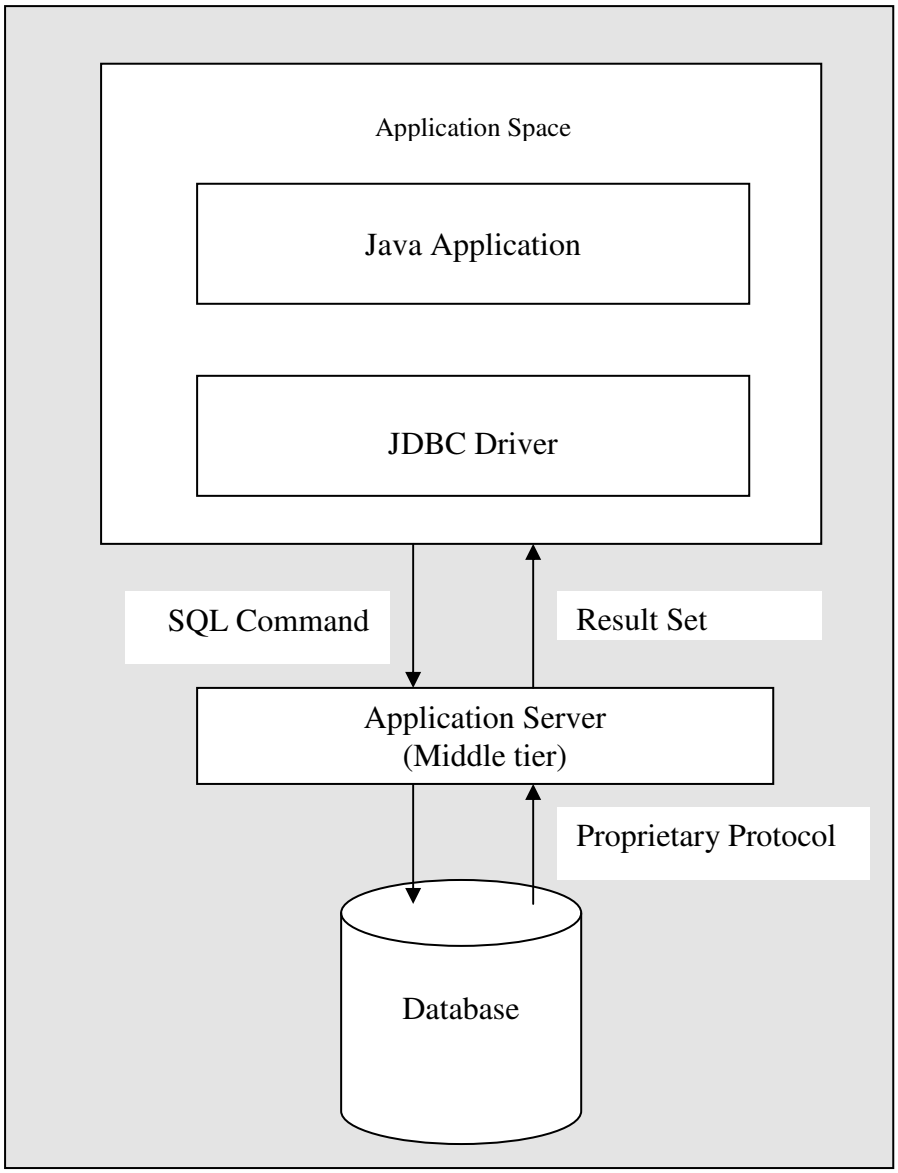


Fig: Three-tier model

3.1.2 ORACLE:

The Oracle database system provides a number of benefits, some of which can also be found in other relational database systems. In addition to the benefits described in the following section, Oracle and it's components run on more than a hundred different hardware and operating system platforms [including all the different versions of UNIX from the many different vendors]. Oracle corporation also provides a full suit of development tools, end-user tools, applications and utilities.

1. Large Database: -

Oracle can support databases ranging in size from megabytes to hundreds of GB. The database file can reside either on hard disk drives or on CD-ROMs, which can be particularly useful for archival or historical data.

2. Many Users: -

Without any extra application development effort, and Oracle databases and support from hundreds of users, the database management software does all necessary locking and protection of data. In fact, it is common to see applications being developed on a single user version of Oracle database and then implemented without any change of code on bigger machine with many more users.

3. Large Range of Tools: -

The Oracle Corporation supplies many tools that provide front-end access to Oracle databases in the form of screens, inputs, reports or even graphs on

the data. These tools though used mostly with Oracle databases, can also be used with non-Oracle data sources. The tool selection expands periodically.

4. Portable: -

The Oracle RDBMS software as already mentioned comes on than hundred different hardware and OS platforms. If one has developed Oracle application on one machine, it becomes relatively easy to put the system to another machine and operating system. Very little needs to be changed with regards to Oracle application code.

5. Backups and Recovery: -

Oracle provides many options when it comes to backing up and recovering the data. In fact, if a database and machine has been setup correctly there is very little that can cause loss of data. If something does go wrong, the depending on the type of failure, the database administrator also needs to do very little to recover the changes saved since the database was last running, ensuring a speedy return into normal work. Oracle can perform hot backups of database file also, which means that the backup can be performed while the file are in use. This feature enables the system to be available 24-hours a day.

6. Distributed Database: -

Oracle enables data physically located in various databases, which may even mean different machines on scattered locations to be treated as one logical database. The physical structure is hidden from the application program.

The fact that the data does not reside in the database to which they are connected is transparent to the application programs as far as locking, read consistency, and rolling back of work is concerned, everything is handled automatically by the database system software. Again, as far as the application developer is concerned, the program behaves as if all data were physically located in local database.

7. Security: -

Standard Oracle database software provides many security facilities, including controlling access to the database, determining a command that can be run, limiting the amount of sources that can be used by individual processes, and defining the label of access to data in the database. If the security privileges of Standard Oracle are not enough [it will be for the vast major cases], the trust option provides even more privileges.

8. Database Data Processing: -

Using the constraints and database trigger facilities of Oracle, much validation and other processing can be defined when tables are first created. This allows one-setup business rules as he setup data structures. The constraints and database trigger remote the need, in theory, for validation and similar processing to be performed in front-end programs.

3.2 Software & Hardware Requirement Specification

The requirement analysis task is a process of discovery, refinement, modeling and specification. This phase is also known as feasibility study.

In this phase, the development team visits the customer and studies their system. They investigate the need of possible software automation in the given system. By the end of the feasibility study, they furnish a document that holds the personnel assignment, cost, project schedule, and target dates. The requirements gathering processes in intensified and focused specially on software. To understand the nature of the program(s) to be built, the system engineer ("Analyst") must understand the information domain for the software, as well as required function, behavior, performance and interfacing. The essential purpose of this phase is to find the need and to define the problem that needs to be solved.

3. 2.1 Computer Based System:

The word "System" is possibly the most *overused* and *abused* term in *technical lexicon*. A computer-based system can be defined as: *a set of arrangement of elements that are organized to accomplish some method, procedure or control by processing information.*

A computer-based system often includes the following elements:

- ***Software:*** Computer programs, data structures and related documentations those serve to affect the logical method, procedure or control that is required.

- ***Hardware:*** Electronic devices that provide computing capability, the interconnectivity devices that enable the flow of data and electronic devices that provide external world function.
- ***People:*** Users and operators of the hardware and software, categorized in two groups: Administrator and User.
- ***Database:*** A large, organized collection of information that is accessed via software.
- ***Documentation:*** Descriptive information that portrays the use and / or operation of the system.
- ***Procedure:*** The steps that define the specific use of each system element or the procedural context in which the system resides.

3.2.2 Software and Hardware Requirements:

The software requirements of the project are as follows:

1. Software : Java 1.3 or more, Swing.
2. RDBMS : Oracle 9i
3. Operating System: Windows 9x

The hardware requirements of the project are as follows:

1. Processor : Intel Pentium III
2. Mother Board : Any
3. RAM : 256 MB, DDR
4. Monitor : Any
5. Keyboard : Any
6. Mouse : Any

3.3 System Analysis:

The term “Analysis” specifies *what* the system should do. Software requirement engineering is a process of discovery, refinement, modeling and specification. Requirements engineering are the systematic use of proven principles, techniques, languages and tools for the coast effective analysis, documentation and ongoing evolution of user needs and the specification of the external behaviour of a system to satisfy those needs.

Requirement analysis is a software engineering task that bridges the gap between system level software requirement engineering and software design.

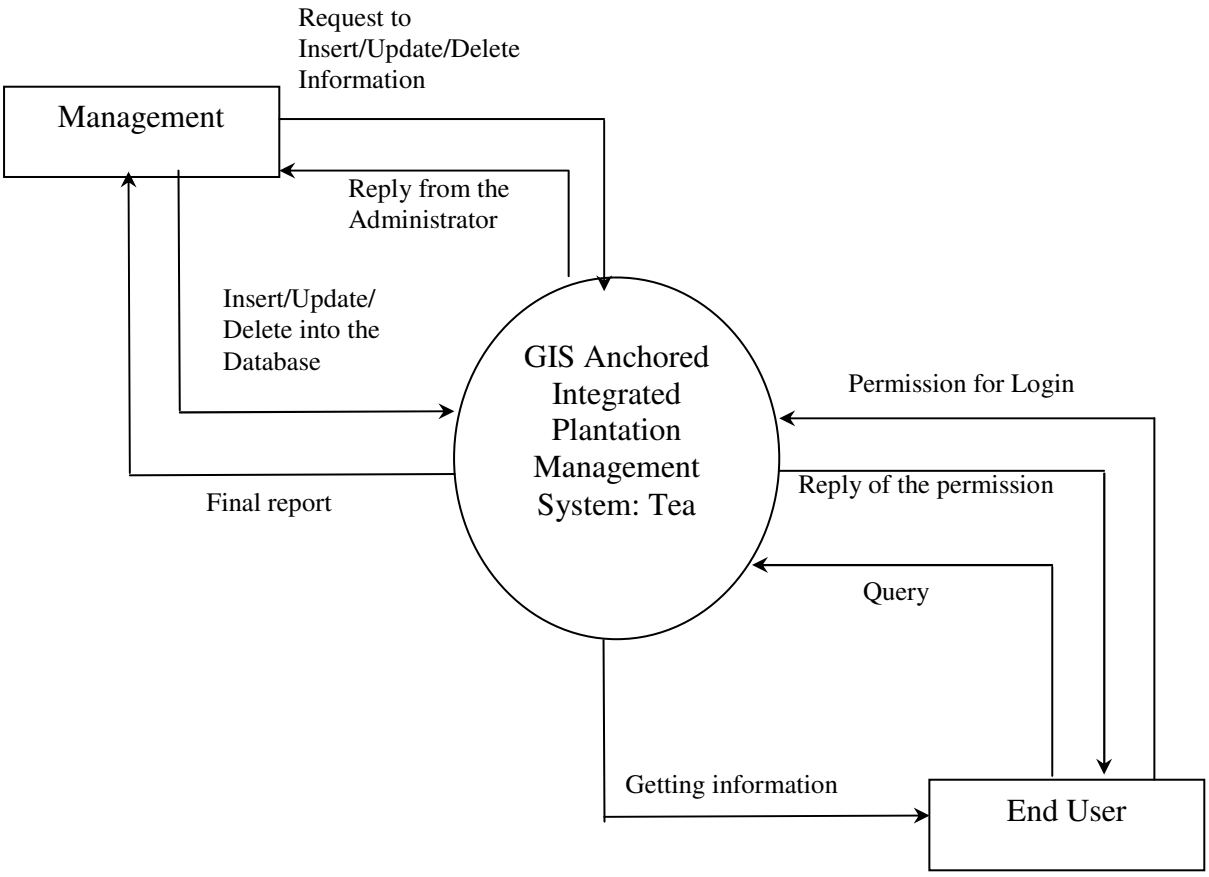
The identification of need preliminary investigation is required to develop the model for the software. The proposed model will be the foundation for completion of the project successfully. A set of elements that are organized to accomplish some predefined goals those will make the successful, by processing information.

Software requirements analysis may be divided into five areas of efforts —

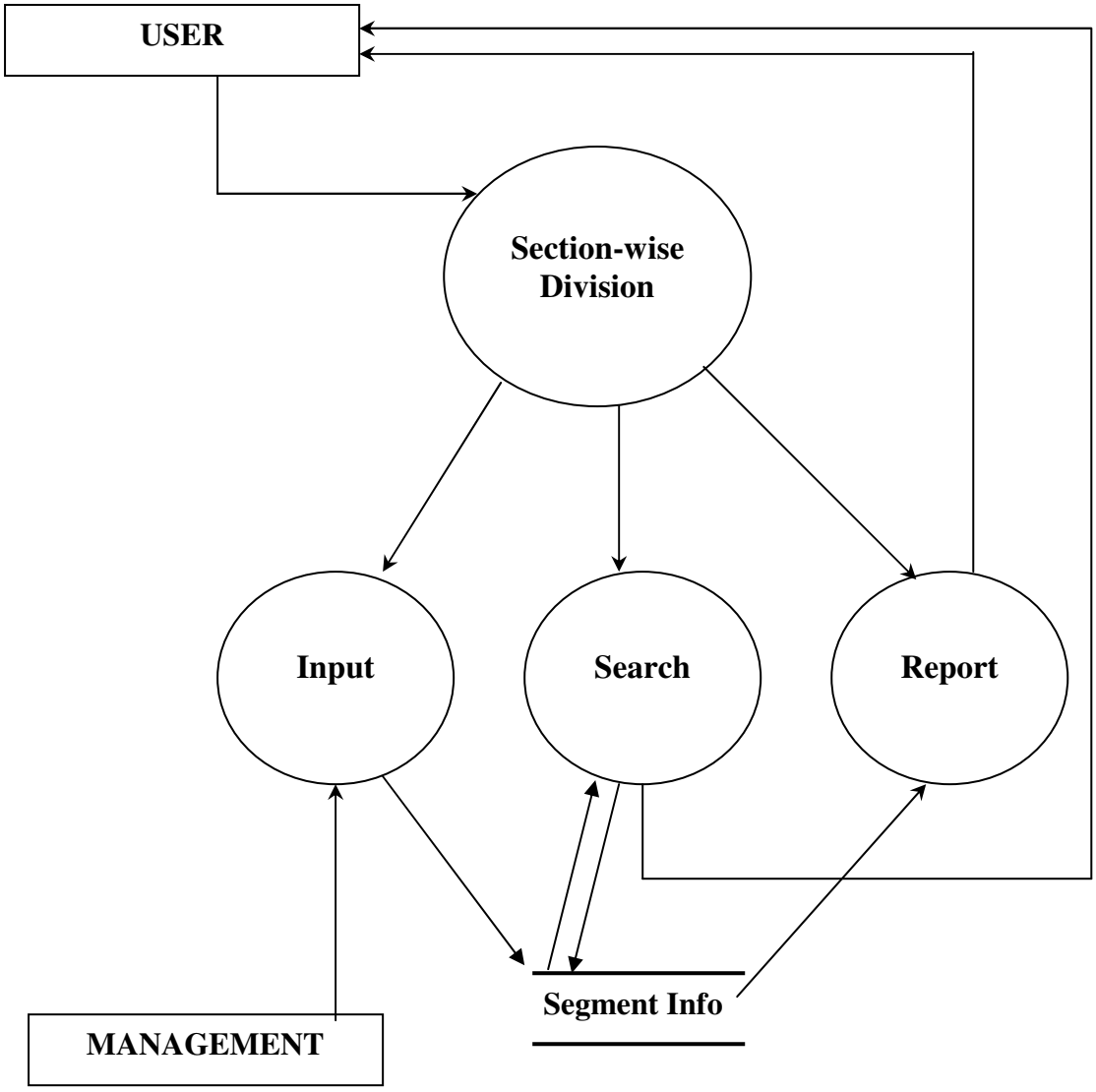
1. Problem recognition
2. Evaluation and synthesis
3. Modeling
4. Specification and
5. Review.

3.4 Schematic Diagram of the System:

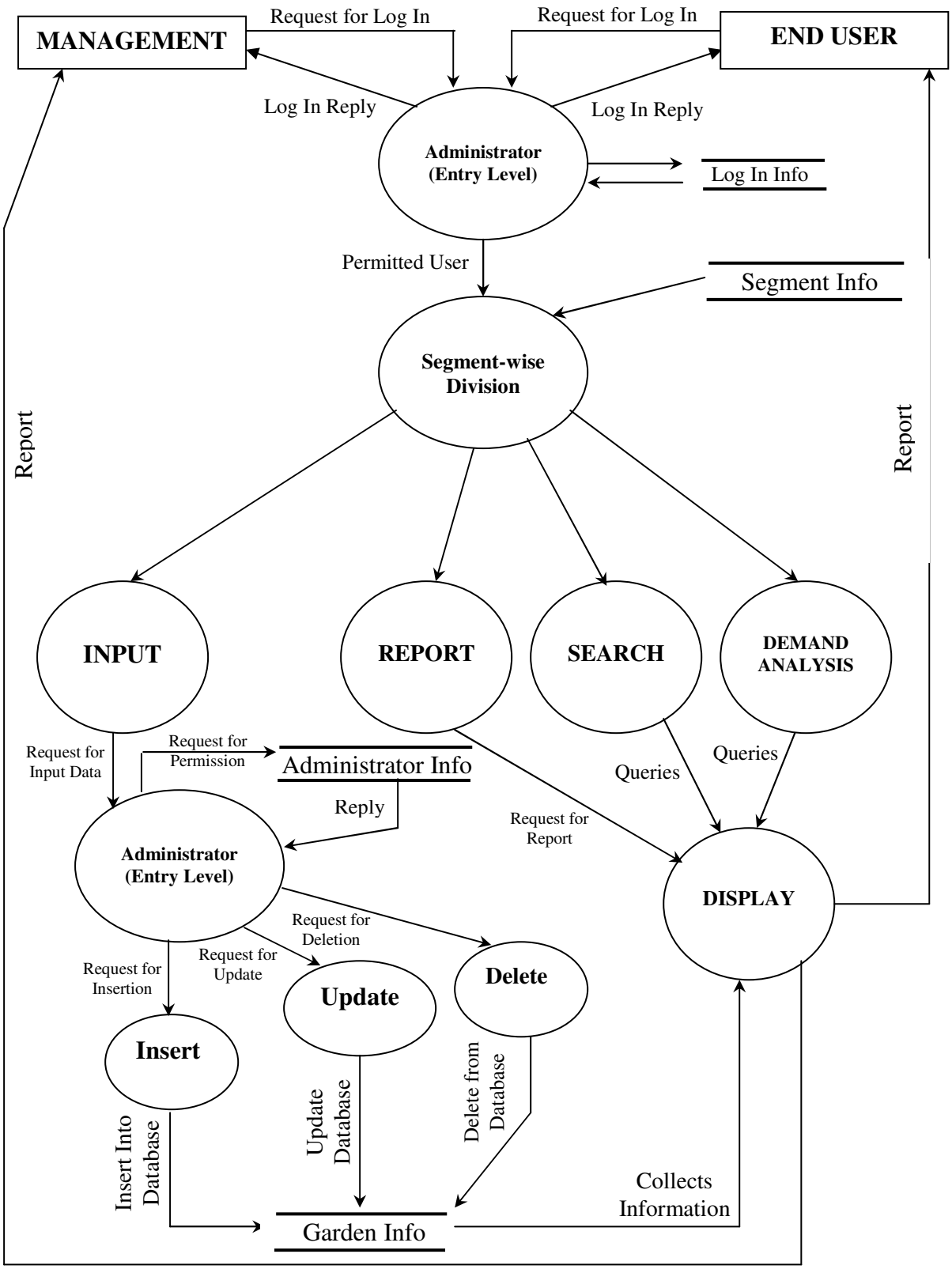
Level 0 DFD



Level 1 DFD


USER
Section-wise Division
Input
Search
Report
Segment Info
MANAGEMENT

Level 2 DFD


MANAGEMENT
Request for Log In
Request for Log In
END USER
Log In Reply
Log In Reply
Administrator (Entry Level)
Log In Info
Permitted User
Segment Info
Segment-wise Division
Report
Report
INPUT
REPORT
SEARCH
DEMAND ANALYSIS
Request for Input Data
Request for Permission
Administrator Info
Reply
Queries
Queries
Request for Report
Administrator (Entry Level)
DISPLAY
Request for Deletion
Request for Update
Request for Insertion
Update
Delete
Insert
Update Database
Delete from Database
Insert Into Database
Collects Information
Garden Info

3.5 Steps to Use the System:

- **Display Image** (“front.jpg”) // front page
- **Sound Play** (“spacemusic.au”)

If (Mouse_Clicked**)**

- Display Image (“p2.jpg”)
- Sound Play (“flute + hrn + mrnba.aif”)

If (Mouse_Clicked)

- Display Image (“12.jpg”)
- Get_User_Name (Check the entered user_id with the administrator_id)
- Get_Password (Check the entered password with the administrator_password)

If (Go_Button Pressed)

- Display (Main screen with original .gif image)

If (Total_Report_Button Pressed)

- Display (The option buttons with the original .gif image)

If (Plucking_Button Pressed)

- Display (Total plucking information)

Else if (Pruning_Button Pressed)

- Display (Total pruning information)

Else if (Manuring_Button Pressed)

- Display (Total manuring information)

Else if (Spraying_Button Pressed)

- Display (Total spraying information)

Else if (Employee_Button Pressed)

- Display (Total employee's information)

Else if (Worker_Button Pressed)

- Display (Total worker's information)

Else if (Back Button Pressed)

- Return (Back to the Main Screen)

If (Audio Help Button Pressed)

- Sound Play ("help.wav")

If (Mouse Clicked upon a segment)

- Display the Section-wise Page

If (Right Mouse Button Pressed)

- Popup Menu Bar (Containing all options such as Plucking, Pruning etc.)
- Get option (Choose the particular option by mouse move)

If (Input option Selected)

- Popup Menu Bar (Containing options Insert, Update, Delete)

If (Insert Menu Selected)

- Get User Name (Get particular user name from combo box)
- Get Password (Enter the user level Password)
- Display (Insert Screen of a Section)
- Get Data (Enter data for the corresponding field)

If (Submit Button Pressed)

- Put Value (Put the value into the Database)

Else if (Back Button Pressed)

- Return (Back to the Main Screen)

If (Update Menu Selected)

- Get User Name (Get particular user name from combo box)
- Get Password (Enter the user level Password)
- Display (Update Screen of a Section)
- Update Data (Update data for the corresponding field)

If (Update Button Pressed)

- Update Value (Update the value of the Database)

Else if (Back Button Pressed)

- Return (Back to the Main Screen)

If (Delete Menu Selected)

- Get User Name (Get particular user name from combo box)
- Get Password (Enter the user level Password)
- Display (Dialog Box of Delete Confirmation)

If (Yes Button is pressed)

- Delete Data (Delete corresponding value from the Database)
- Return (Back to the Main Screen)

Else

- Return (Back to the Main Screen)

If (Report Menu Selected)

- Popup Menu Bar (Containing all options such as Plucking, Pruning etc.)

If (Plucking Menu Selected)

- Display (Plucking Information of that Section)

If (Pruning Menu Selected)

- Display (Pruning Information of that Section)

If (Manuring Menu Selected)

- Display (Manuring Information of that Section)

If (Spraying Menu Selected)

- Display (Spraying Information of that Section)

If (Employee Menu Selected)

- Display (Employee Information)

If (Worker Menu Selected)

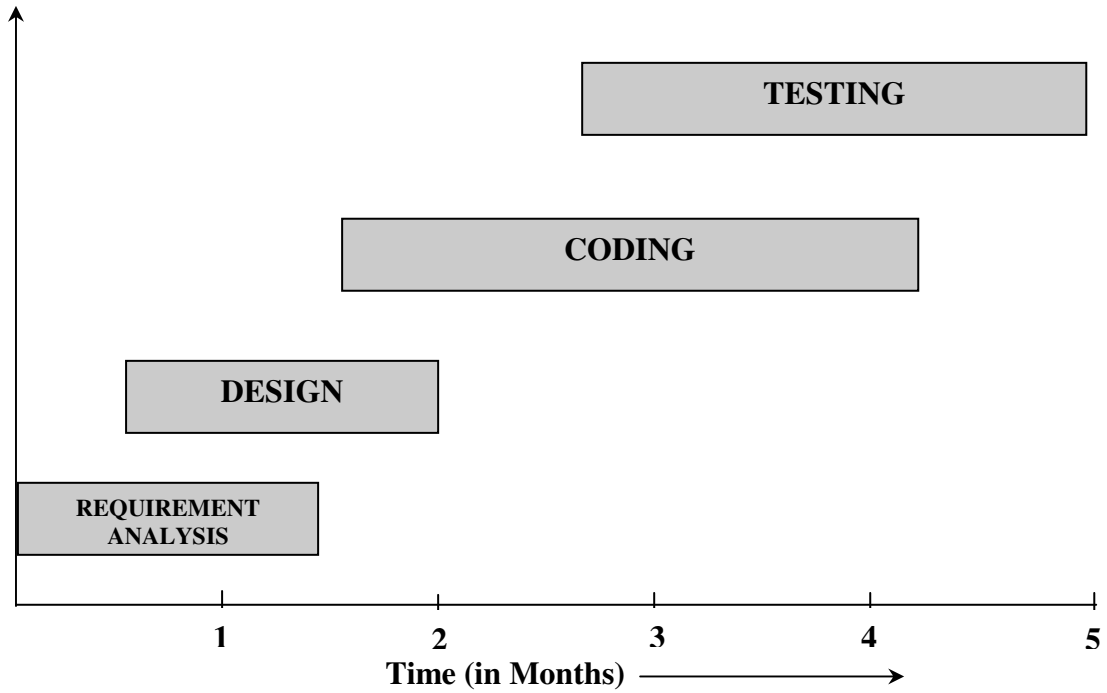
- Display (Worker Information)

If (Search Menu Selected)

- Enter Query (Enter the query)
- Get Result (Get the corresponding Information from the Database)

3.6 Time Scheduling of the Project Work

Total Time: Five Months (5th January, 2006 to 24th May, 2006)



4

Design of the Software

Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built.

Computer software design changes continually as new methods, better analysis, and broader understanding evolves. Software design sits at the kernel of the software engineering process and is applied regardless of the development paradigm that is used.

4.1 Object Oriented Design:

Object-Oriented Design (OOD) creates a representation of the real-world problem domain and maps it into a solution domain that is software. In OOD, an object defines every real-world entity. Every object contains a set of data structure that represent the object's property and a set of methods to work upon them.

4.2 Database Design:

The software uses the following tables for storing information regarding the Tea Garden project:

1. AREA
2. PRUNING
3. PLUCKING
4. SPRAYING
5. MANURING
6. W_MASTER
7. EMP_MASTER

The details of the tables' contents are:

AREA Table:
(Contains information about area of each section of the tea garden)

Field Name	Data Type	Null?	Description
SEC_CODE	VARCHAR2(10)	Not Null	Unique Code of the section
SEC_AREA	NUMBER(5,2)		Area of the corresponding section

PRUNING Table:
(Contains information about pruning of each section of the tea garden)

Field Name	Data Type	Null?	Description
SEC_CODE	VARCHAR2(10)	Not Null	Unique Code of the section
L_TYPE	VARCHAR2(10)		Last pruning type
L_DATE	VARCHAR2(15)		Last pruning date
N_O_W	NUMBER(5)		No of workers for last pruning
N_TYPE	VARCHAR2(10)		Next pruning type
N_DATE	VARCHAR2(15)		Next pruning date

PLUCKING Table:
(Contains information about plucking of each section of the tea garden)

Field Name	Data Type	Null?	Description
SEC_CODE	VARCHAR2(10)	Not Null	Unique Code of the section
PLUCK_DATE	VARCHAR2(15)		Current plucking date
T_TOT_YIELD	NUMBER(10,2)		Total yield of the corresponding date
N_O_W	NUMBER(5)		No of workers for plucking
M_YIELD	NUMBER(10,2)		Current month's total yield
Y_YIELD	NUMBER(10,2)		Current year's total yield

MANURING Table:
(Contains information about manuring of each section of the tea garden)

Field Name	Data Type	Null?	Description
SEC_CODE	VARCHAR2(10)	Not Null	Unique Code of the section
L_M_DATE	VARCHAR2(15)		Last manuring date
N_O_W	NUMBER(5)		No of workers for plucking
N_M_DATE	VARCHAR2(15)		Next manuring date

SPRAYING Table:
(Contains information about spraying of each section of the tea garden)

Field Name	Data Type	Null?	Description
SEC_CODE	VARCHAR2(10)	Not Null	Unique Code of the section
L_S_DATE	VARCHAR2(15)		Last spraying date
N_O_W	NUMBER(5)		No of workers for plucking
N_S_DATE	VARCHAR2(15)		Next spraying date

W_MASTER Table:
(Contains information about workers of the tea garden)

Field Name	Data Type	Null?	Description
W_ID	VARCHAR2(5)	Not Null	Unique code for each worker
W_NAME	VARCHAR2(30)		Name of the worker
W_ADD	VARCHAR2(40)		Address of the worker
W_SEX	VARCHAR2(15)		Sex of the worker
W_TYPE	VARCHAR2(10)		Type of the worker
W_YR_YIELD	NUMBER(10,3)		Total yield of a worker in a year

EMP_MASTER Table:
(Contains information about employees of the tea garden)

Field Name	Data Type	Null?	Description
EMP_ID	VARCHAR2(8)	Not Null	Unique Code for each employee
NAME	VARCHAR2(25)		Name of the employee
ADDRESS	VARCHAR2(30)		Address of the employee
SEX	VARCHAR2(6)		Sex of the employee
D_O_B	VARCHAR2(12)		Date of birth of the employee
D_O_J	VARCHAR2(12)		Date of joining of the employee
DSGN	VARCHAR2(20)		Designation of the employee
BASIC	NUMBER(8,2)		Basic Salary of the employee
DA	NUMBER(7,2)		Dearness Allowance of the employee
HRA	NUMBER(7, 2)		House Rent Allowance of the employee
SA	NUMBER(7, 2)		Special Allowance of the employee
PF	NUMBER(7, 2)		Provident fund of the employee

4.3 User Interface Design

The User Interface Design is the mechanism through which a dialog between the program and the human is established. A simple and user-friendly user interface is worth in any software. The interface should be such that any user can handle the software effortlessly.

The following are few forms those the user has to go through while handling the software:

Fig.: - Window shows the Front Page of the Package

Fig.: - Window shows Brief Details of Project Guide & Trainees.

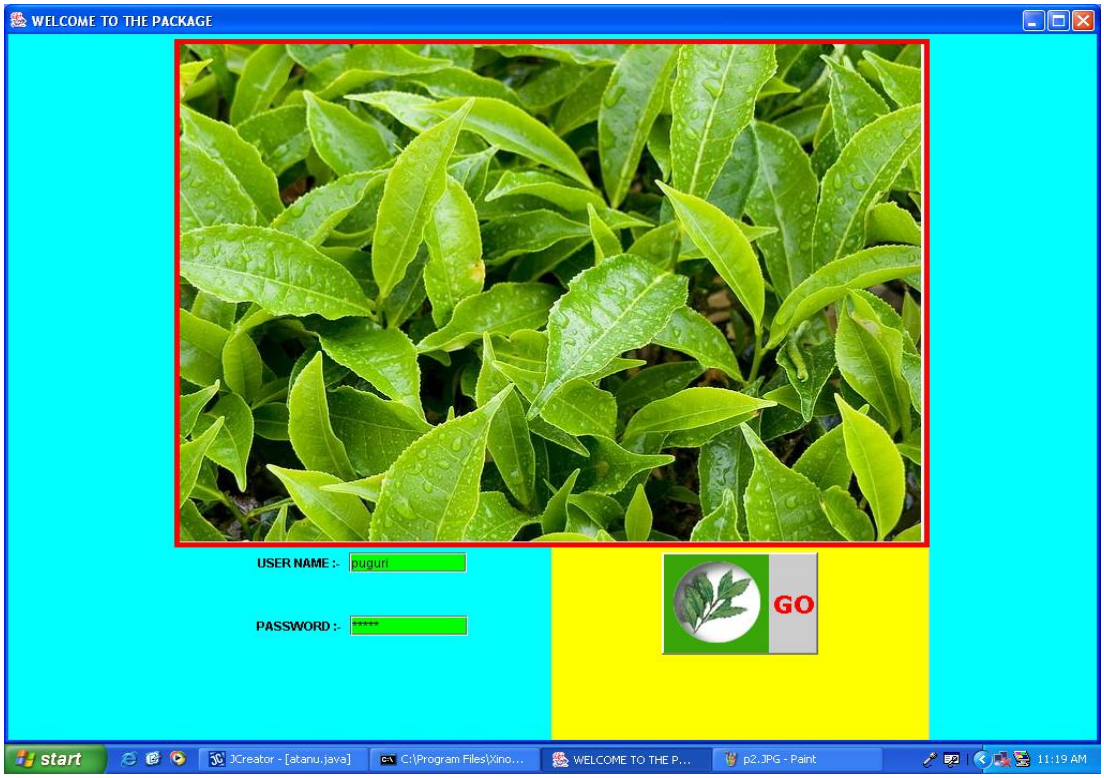


Fig.: - Window shows the Administrator Level Password Checking.

Fig.: - Window shows the Main form where Zoomed View of the corresponding Section where mouse is over in the main map.

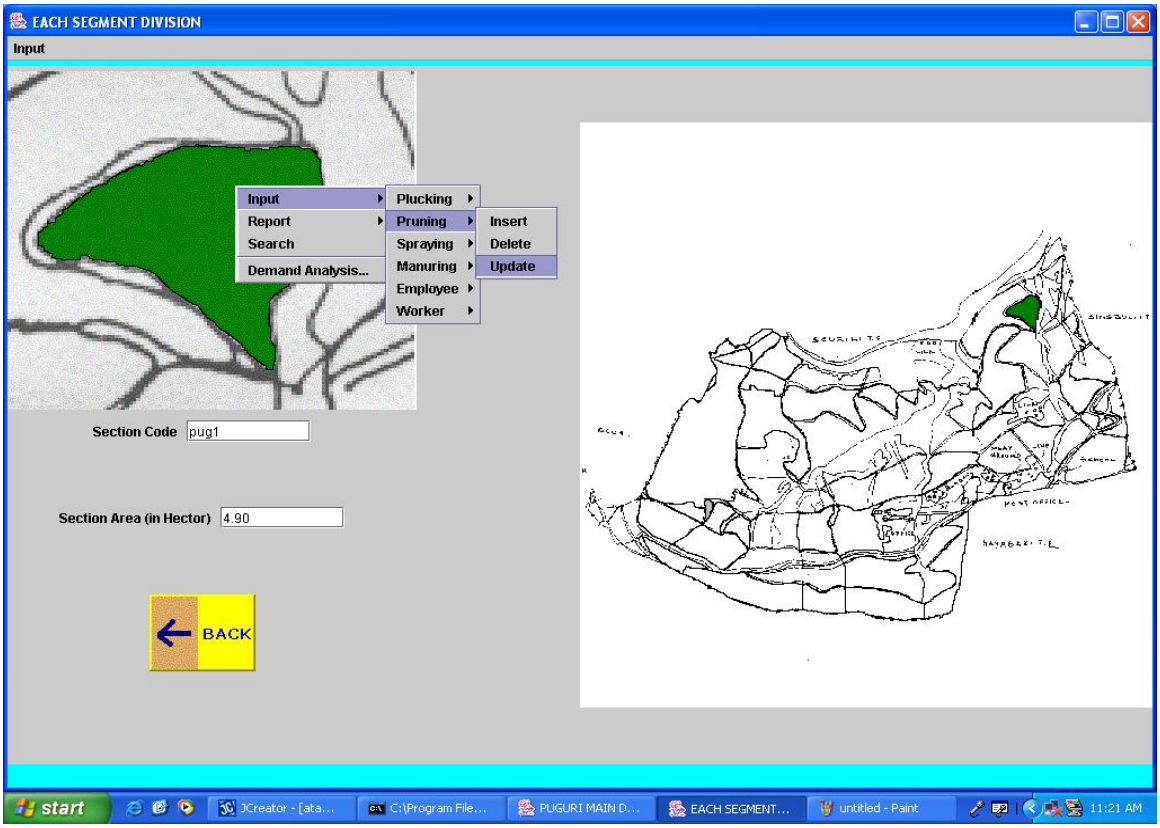


Fig.: - Window shows the Popup Menu for the selected Section

5

Results

So far, the system has been generated and some experiments have been carried out to the system. Few such experiments and corresponding results are listed below.

5.1 Insertion of the Data:

After selecting a specific section for work, when right clicked into the section, the pop up menu will appear. In the pop up menu, when the Pruning/Plucking/Manuring/Spraying/Employee or Worker menu item is clicked, **Insert** sub-menu item appears. Now when **Insert** sub-menu is clicked, an administrator username & password is checked. If the user is authorized then only he/she can insert the data for that corresponding section.

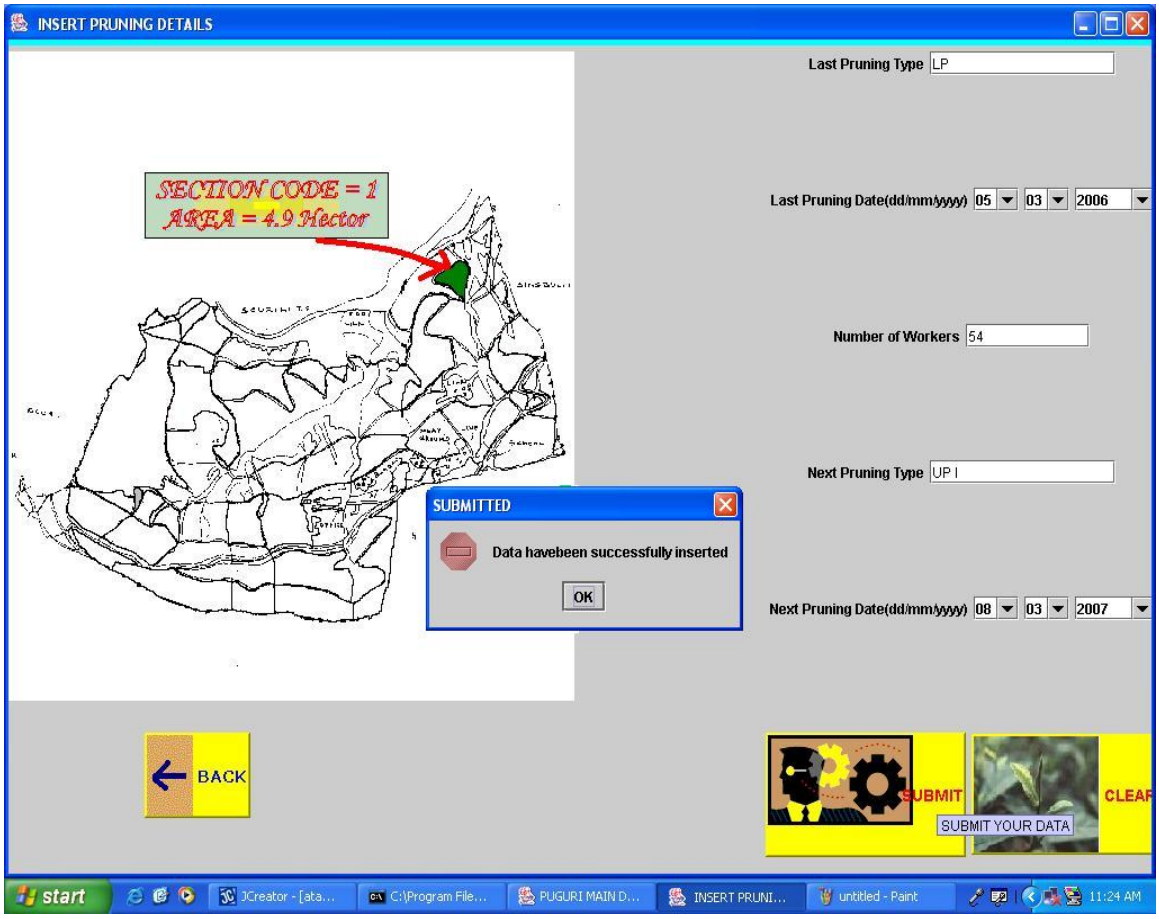


Fig: - Window shows the Form for the Insertion of Data

5.2 Update of the Inserted Data:

After selecting a specific section for work, when right clicked into the section, the pop up menu will appear. In the pop up menu, when the Pruning/Plucking/Manuring/Spraying/Employee or Worker menu item is clicked, **Update** sub-menu item appears. Now when **Update** sub-menu is clicked, an administrator username & password is checked. If the user is authorized then only he/she can update the data for that corresponding section.

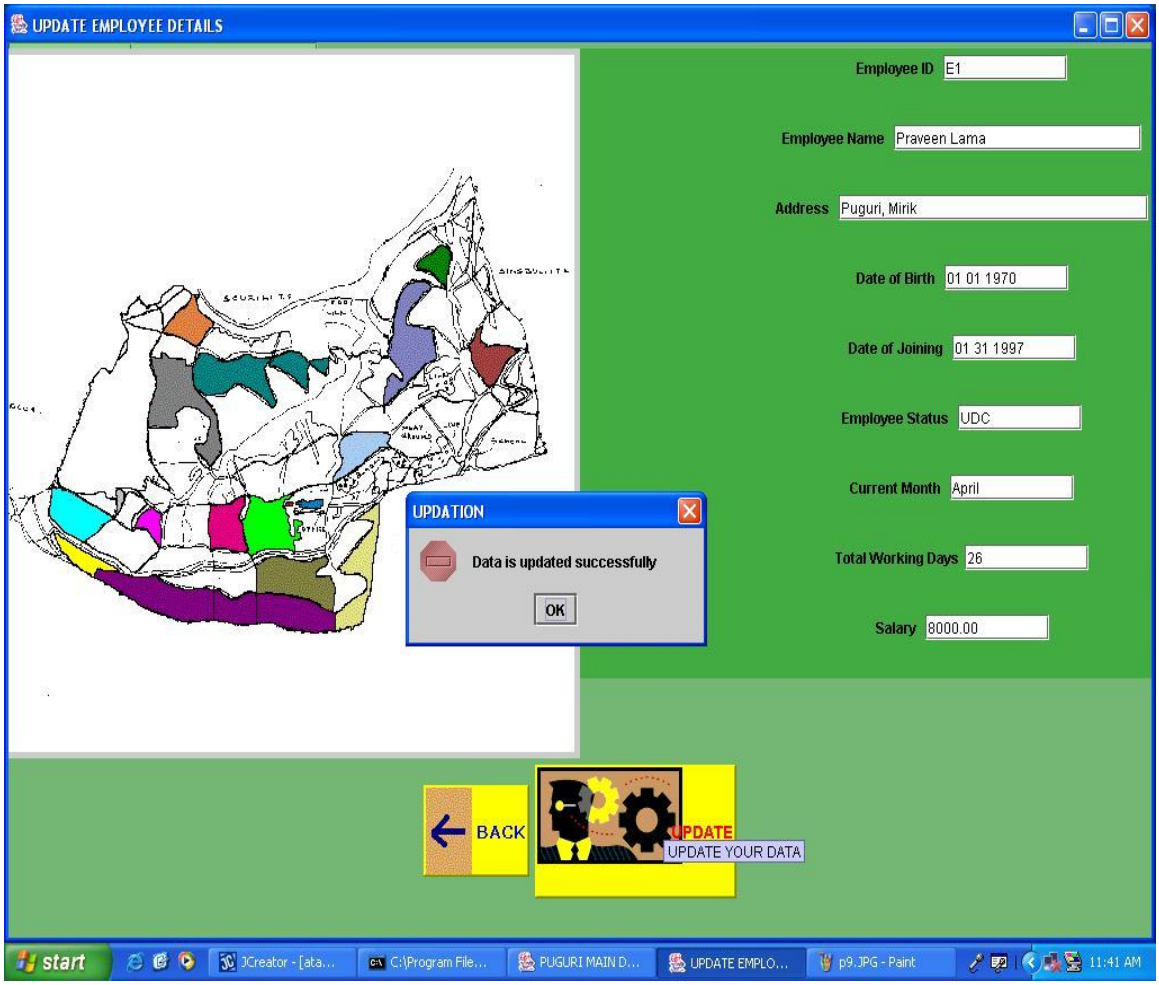


Fig: - Window shows the form for Update of the Inserted Data

5.3 Deletion of the Data

After selecting a specific section for work, when right clicked into the section, the pop up menu will appear. In the pop up menu, when the Pruning/Plucking/Manuring/Spraying/Employee or Worker menu item is clicked, **Delete** sub-menu item appears. Now when **Delete** sub-menu is clicked, an administrator username & password is checked. If the user is authorized then only he/she can delete the data for that corresponding section.

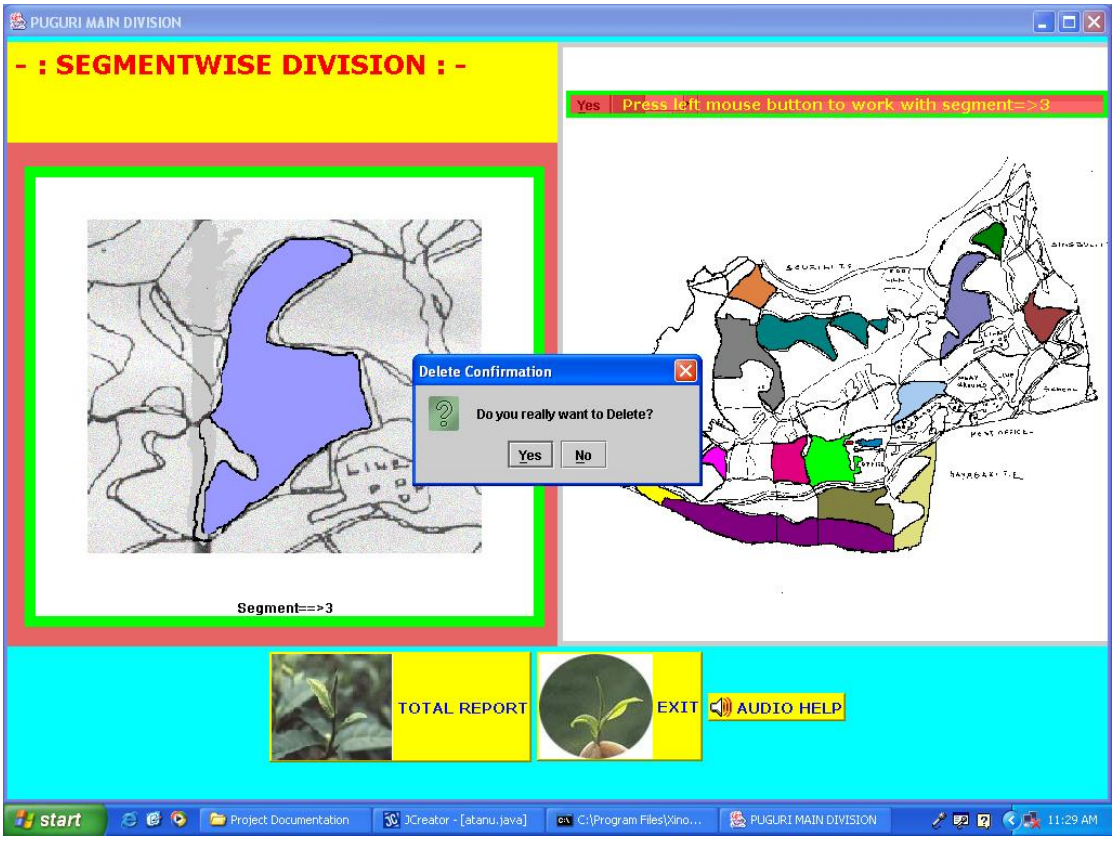


Fig: - Window shows the deletion of the Data for the selected section

5.4 Total Report Generation

Details report of the garden can be viewed from the **Total Report** button of the main form. The form given below shows the different fields of the garden's report.

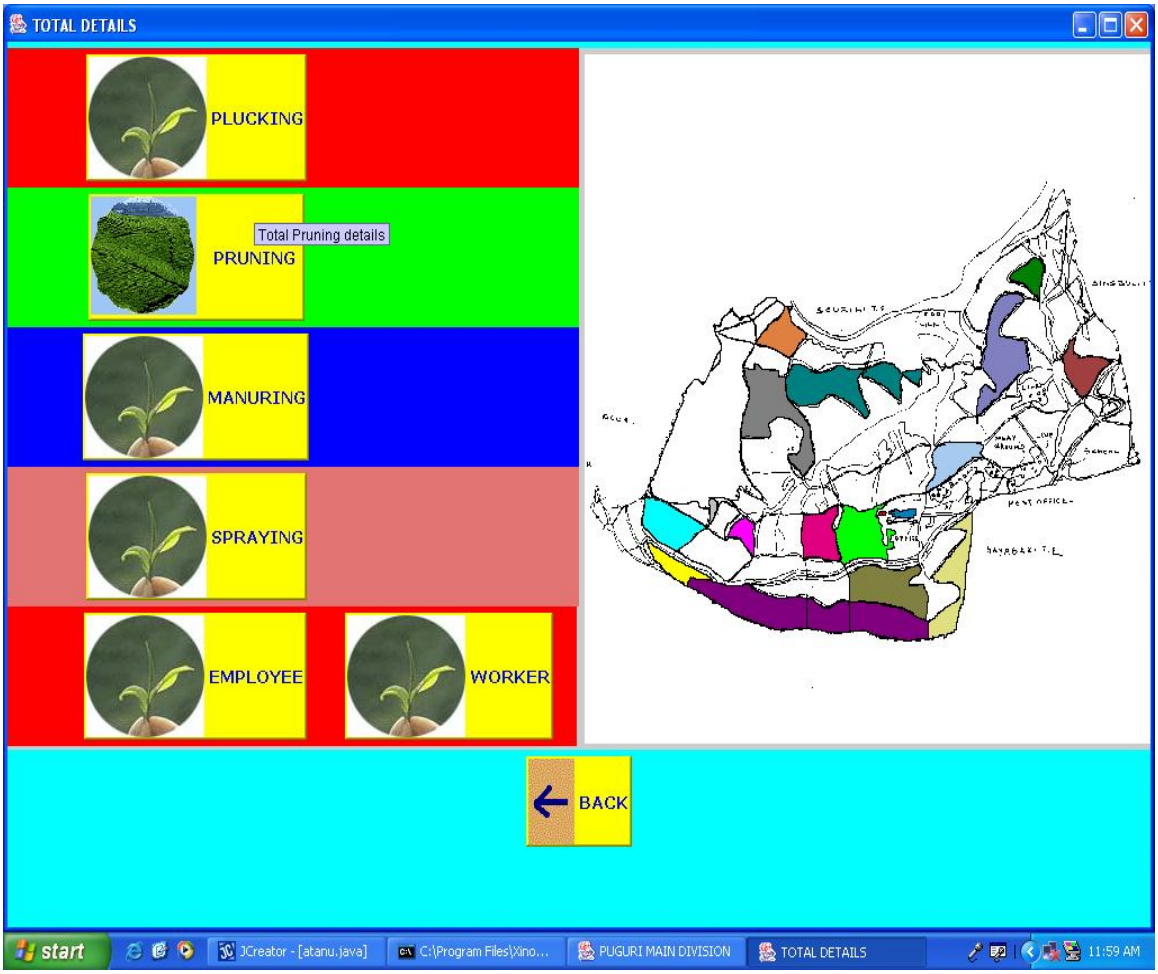


Fig: - Window shows the Details Report form for different fields

5.5 Detail Reports

Details report of the garden can be viewed from the **Total Report** button of the main form. The form given below shows the different fields of the garden's report.

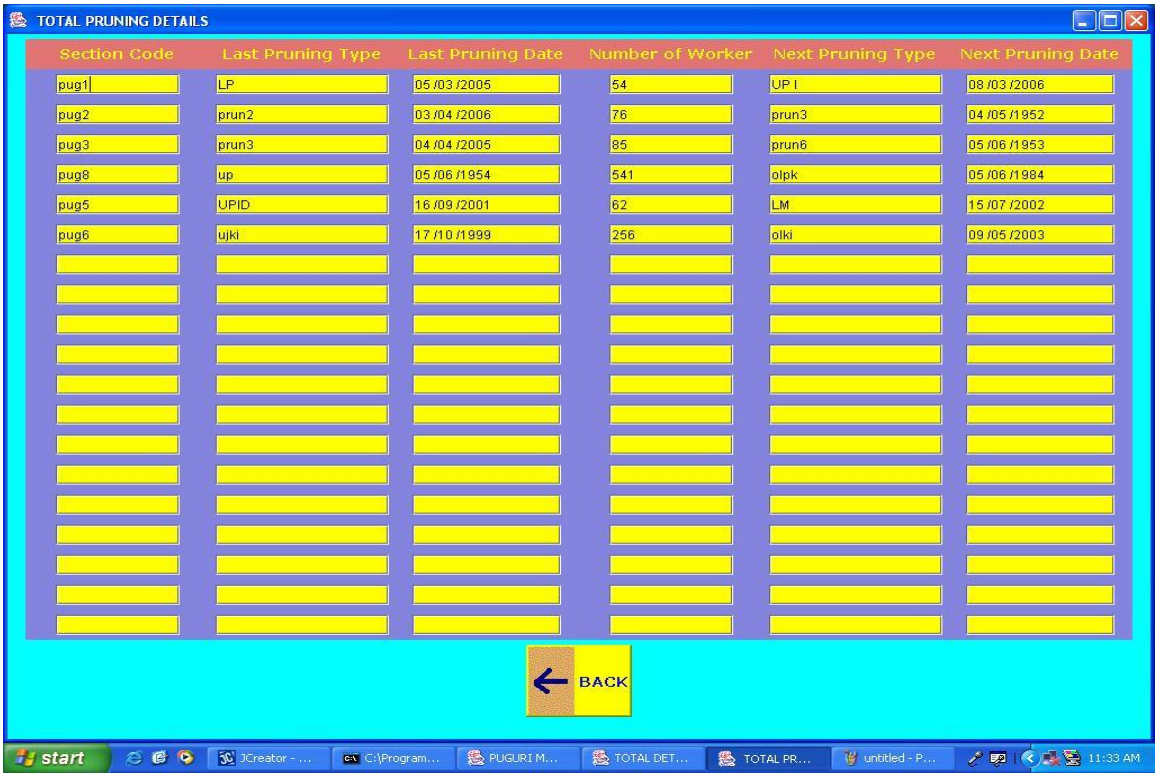


Fig: - Window shows the Total Pruning Details.

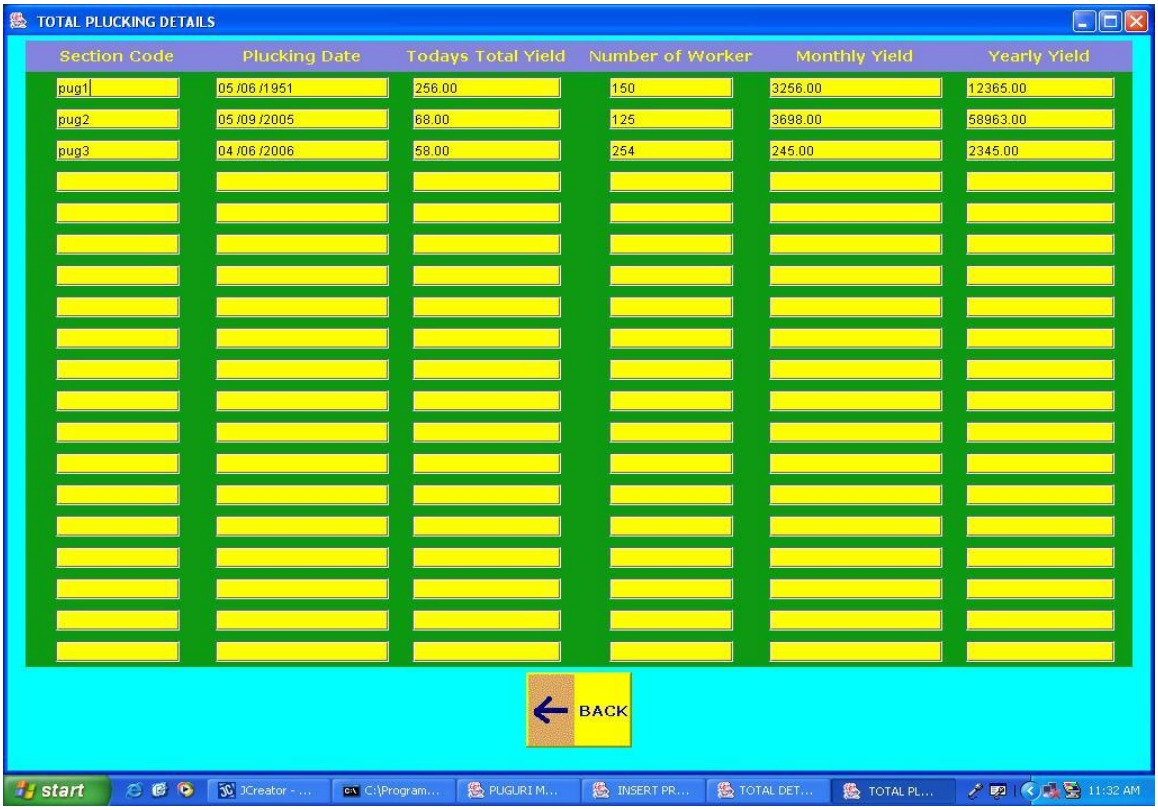


Fig: - Window shows the Total Plucking Details

6

Discussion

The Geographical Information System or GIS, in short, is rapidly a growing technology for representing spatial data all over the world. Interpretation of database information has always been cumbersome and complex. On the other hand, visual data representation is worth a thousand words.

Tea is indigenous to India and is an area where the country can take a lot of pride. This is mainly because of its pre-eminence as a foreign exchange earner. In all aspects of tea production, consumption and export, India has emerged to be the world leader, mainly because it accounts for 31% of global production. It is perhaps the only industry where India has retained its leadership over the last 150 years. The range of tea offered by India, from the original Orthodox to CTC and Green Tea, remains unparalleled in the world. Even then, for the last couple of years, due to rapid changes in global economic scenario, liberalization of Indian economy and increasing presence of countries like Kenya, Indonesia, Sri Lanka etc. in global export market, Indian tea is under recession. At the same time, due to use of too much pesticide, the global market refuses to import Indian tea. Considering all these aspects, a management system for tea gardens is proposed here to enhance the quality and productivity. So the overall objectives of this proposed project is to generate a ***GIS Anchored Integrated Plantation Management: Tea*** to maximize the profit in the tea gardens through the strength of IT in the form of GIS.

Barring some accounting works, almost all the tea gardens in India are being managed manually. Due to large amount of data in tea gardens the task of making management decisions becomes time consuming and tedious and this may be a reason behind the low quality and production rate. To improve the quality and

production of tea, it is necessary to consider the following aspects of tea management–

Plucking scheduling,

Pruning scheduling,

Spraying scheduling,

Scheduling of manuring,

Employee and Labour management.

In addition there is one more important aspect of Tea Management, i.e., Drainage System, which is beyond the scope of the proposed work. Brief descriptions of the aforesaid aspects are mentioned below.

6.1 Plucking of Tea

Plucking in tea cultivation is synonymous with harvesting in other crops. The tender apical portion of the shoot consisting of 2-3 leaves and the terminal bud are nipped off in plucking. The general time of plucking is starting of March to end of November.

Plucking Rounds:

The time interval between two successive plucking is called a ***Plucking Round***. Plucking round may be extended from 4 to 14 days, but to keep a balance between crop and quality, plucking round of 6-8 days are usually practiced. The time required for unfolding of successive leaves from a growing bud varies from 3 to 6 days depending on the climatic variations. This is called a ***Leaf Period***. The mean leaf period of tea bushes of North-East India is 4 days

during the main flushing season and the leaf should not be plucked a day earlier than twice of the leaf period (i.e., 2 x leaf period – 1 = 7 days). An automated management system (database oriented) may be helpful to monitor the plucking of tea.

6.2 Pruning Scheduling:

Pruning is one of the most important operations, next to plucking, which directly determines the productivity of tea bushes. If pruning is delayed, in other words as the age of wood from pruning increases, the size and weight of growing shoots on plucking surface decreases. Therefore to maintain the vegetative growth, pruning is necessary.

Light prune (LP):

Tea bushes are usually pruned every 3 or 4 years at 4- 5 cm above the last pruning cut. The time period from one light prune year to another is called one pruning cycle.

Height reduction prune and Medium prune:

When the tea bushes grows tall and plucking becomes difficult, they are brought down to an optimum height by height reduction prune (HRP) at 60 – 70 cm, or medium prune (MP) at 45 – 60 cm above ground.

In between two successive prune (LP) years, tea bushes are given lighter forms of cuts which are termed as deep skiff (DS), medium skiff (MS), light skiff (LS), level of skiff (LOS) or untouched which is called unpruned.

Time of pruning and skiffing: pruning is done in November to December. Skiffing can be performed at any time between mid-December to mid January.

6.3 Spraying Scheduling:

Pesticide should be sprayed immediately after plucking but foliar nutrients and growth regulators to be sprayed 3 – 4 days after plucking. Pesticide and foliar nutrients better to spray separately. Spraying should have to avoid during high sunshine hours.

Now a day's foreign country avoid buying tea that contains any kind of chemical insecticide. So taking precautions from pests and spraying chemical insecticide in advance is less practiced now. If the pests suffer any area of the garden then only insecticide is sprayed in that very zone.

6.4 Scheduling of Manuring:

It is important to ensure adequate replenishment of nutrients in the soil removed by harvest. In the Darjeeling hills the agro climatic conditions are different from those of plains. The cool temperature slows down the organic matter

decomposition and the productivity levels are lower as compared to the plain districts. Therefore the N levels should not exceed 110 – 120 kg / ha. The Table below shows the recommended NPK Manuring in mature tea in Darjeeling.

Yield Range in KMTH	N kg / ha	P2O5 kg / ha	K2O kg / ha		
			Soil available potash status		
			Low (< 60 ppm)	Medium (60-100 ppm)	High (> 100 ppm)
Up to 600	Up to 60	20	Up to 60	Up to 50	Up to 35
600-1000	60 - 90	20	60 – 90	50 – 70	35 – 50
1000-1400	90 - 120	20	90 – 120	70 – 100	50 - 70

Till the garden can be enriched with organic build up process of inorganic fertilizer application will continue. In many gardens, exact requirement of fertilizer is not ascertained which in turn result in either excess of deficient fertilizer application. Somewhere right kind of fertilizers is not used. In a soil having high pH (above 5.5), rock phosphate should not be used because it is an acid soluble phosphate. Single super Phosphate, which is water soluble Phosphate should be the best proposition in those kind of soil.

Time of manuring:

Fertilizers should be applied only when the tea bushes are ready to utilize them. As a general rule, the best time for fertilizer application is after the first rain in spring has moistened the soil to a depth of 45 cm. In case of pruned and skiffed teas, the fertilizer should be applied after the bushes have produced two new leaves.

It is recommended to apply N and K in two splits if the dose exceeds 100 kg / ha. 60 % should be applied on March – April and 40% of N and K should be applied on August – September.

It is sometimes recommended to add Sulphur @ 20 – 45 kg /ha to improves yield and quality of tea. Sometimes Zinc is beneficial for increase the yield but recommendation is not to exceed 12.5 kg zinc sulphate / ha / yr.

Foliar nutrition is beneficial under stress condition or coinciding with physiological changes in the bush. NPK mixture 2-1-2 or 2-1-3 where potash status is low, @0.5-1 percent can be sprayed during this period.

Use of bio-fertilizer viz. Azotobacter, Phosphobactrin, Azospirrilium, Bacillus megatherium, can be quite productive.

6.5 Types of Workers:

There are generally 6 types of workers in any tea garden. They are all given the facilities like gratuity, bonus, provident fund, ration, free house, fire wood and compulsory medical treatment and tea except the casual workers. These casual workers are called for work when ever they necessary, so they are also called seasonal workers. The types of workers are ——

1. Daily Rated Workers (Paid forth nightly)
2. Supervisors or Sub-staffs (Monthly rated)
3. Clerical staffs (Monthly rated)
4. Technical staffs (Monthly rated)
5. Medical Staffs (Monthly rated)
6. Casual or Seasonal Workers (Daily rated)

7

Field Study

To get a sharp knowledge about the total system for the completion and betterment of the project we, the team members of the project visited the work field Phuguri Tea Estate, Mirik, Darjeeling two times during the continuation of the project work. The first visit of the work field by us was on 30th November, 2005 to 1st December, 2005 and the second was on 17th April, 2006.

During these visits the team members met with the General Manager of the Tea Estate and collected various required information for the development of the entire system. We collected the information regarding various pruning types and pruning, plucking, manuring, spraying scheduling. There we got an overall view of the tea garden management that helped us in every step to develop the package.

We have also collected data of various workers and employees of the tea estate. These data contains details of the facilities that the workers and employees of the garden get from the management. This also helped us to create the worker and permanent employee databases.

8

Future Scope

Tea being one of the most important and highest foreign exchange earning agricultural products of India is now under recession. Industry and Government has realized that there is immediate need to modernize the tea production process to make it globally competitive. The major national objective of tea industry is to utilize available land on estates towards sustainable development by ensuring land use efficiency with proper soil conservation measures and environment protection. In addition, producers' objective to is to manage tea estates to maximize profit.

In our project, we worked for a specific tea estate namely Phuguri Tea Estate, Mirik, Darjeeling. In future the package will be a web-based GIS integrated package which can be accessed from any platform. A part of the package is still to be developed under Linux platform.

Our work in the project is just a prototype of the entire package which is still to be developed for the Tea Industry of India. The accumulated knowledge, data and information of our project can easily be used in future days in the advanced fields of Information Technology and Geographical Information System based development. As a result, this will have penetration prospect also in other similar plantation based industries like Coffee.

The end product of the entire project will be a customizable GIS-ERP package which will be beneficial to the Tea-Planters in terms of better planning, management and enhanced productivity and profitability.

9

Coding

To translate representation of the software into a form that is "understood by the computer, we need a programming language. Coding is the process that transforms the design into programming language. Programming language characteristics and coding style can profoundly affect software quality and maintainability.

9.1 Language Chosen:

Given a variety of available programming languages e.g. C, C++, PASCAL, BASIC, JAVA etc. the following criteria should be there by choosing the programming language — (i) general application area, (ii) algorithmic and computational complexity, (iii) environment in which the software will execute, (iv) data structure complexity, (v) performance consideration, (vi) knowledge of software development staffs, and (vii) availability of good compiler or cross compiler.

The programming language chosen for the development of the software is **JAVA**. Java satisfies all the criteria stated above.

The basic criterion for the software requires that it is to be operated over any computer and the data should be secured. Java is the best available object -oriented language. It is platform independent.

9.2 Coding Style:

After source code is generated the function of a module should be apparent without reference to a design specification, that is, the code must be understandable.

9.2.1 Code Documentation:

Internal documentation of source codes begins with the selection of identifier (variables and labels) names, continues with the placement and composition of commenting, and concludes with the visual organization of the program.

The variables declared in the program are given meaningful names as mush as possible. For example, an object of the class MainForm given the name mainForm, so that one can readily understand that mainForm is an object of the MainForm.

The code are well commented. The comments are written using standard Java comments e.g. //, /* and */. The codes are also well indented with inner codes having higher indentation.

9.2.2 Data Declaration:

While the declaration of data, the following order has been followed:

1. All explicit declaration.
2. All global data
3. All local Arrays
4. All file declaration

Also while declaring multiple variable names in a single statement, the alphabetical ordering of the name has been followed.

9.2.3 Statement Construction:

The statements have been written in multiple formats, with each statement one line and proper indentation.

9.2.3 Input/ Output:

The following input/output style has been chosen while coding the software.

(i) Validation Checking of all input data

(ii) All input data are kept simple as far as possible.

9.3 Selected Codes:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.util.*;
import java.sql.*;
import java.applet.*;
import java.awt.image.*;
import java.awt.Graphics.*;
import javax.swing.event.*;
import java.applet.AudioClip;
import java.net.URL;
import java.net.MalformedURLException;
import java.awt.GridBagLayout;
```

//Code for Front Page

```
public class gisproject extends Frame implements MouseListener
{
        JFrame mf;
        JLabel lab;
        Container c;
        int flag=0;

   SoundList soundList;
   String auFile = "spacemusic.au";
   String aiffFile = "flute+hrn+mrmba.aif";
   String midiFile = "trippygaia1.mid";
   String rmfFile = "jungle.rmf";
   String wavFile = "bottle-open.wav";
   String chosenFile;

   AudioClip onceClip, loopClip;
```

```
URL codeBase;


boolean looping = false;

      gisproject ()
      {
          lab=new JLabel(new ImageIcon("D://image/front.jpg"), JLabel.LEFT);
               mf=new JFrame("WELCOME TO THE PACKAGE");
               mf.setSize(1024,738);
               JPanel p=new JPanel();
               p.setLayout(new GridLayout(2,1));
               p.addMouseListener(this);
               p.add(lab);
               c=mf.getContentPane();
               c.add(p);
               sound(1);
   c.setLayout(new FlowLayout());
      mf.setVisible(true);

      }

public void sound(int k)
{

   chosenFile = auFile;
   addMouseListener(this);
   startLoadingSounds();
}

 void startLoadingSounds()
{

   try {
      codeBase = new URL("file:" + System.getProperty("user.dir") + "/");
   } catch (MalformedURLException e) {
      System.err.println(e.getMessage());
   }
   soundList = new SoundList(codeBase);
   soundList.startLoading(auFile);
   soundList.startLoading(aiffFile);
   soundList.startLoading(midiFile);
   soundList.startLoading(rmfFile);
   soundList.startLoading(wavFile);

}

public void stop() {
```

```
        onceClip.stop();        //Cut short the one-time sound.
        if (looping) {
           loopClip.stop();    //Stop the sound loop.
        }
    }

    public void start() {
        if (looping) {
           loopClip.loop();    //Restart the sound loop.
        }
    }
public static void main(String args[])
{
          new gisproject ();
}

       public void mouseClicked(MouseEvent me)
       {
             if (looping)
             {
         looping = false;
         loopClip.stop();    //Stop the sound loop.

        }
        else if (onceClip != null)
         {
              onceClip.stop();
         }
          return;
           }
           public void mouseDragged(MouseEvent me)
           {

           }
           public void mouseEntered(MouseEvent me)
           {
             onceClip = soundList.getClip(chosenFile);

       onceClip.play();     //Play it once.

       loopClip = soundList.getClip(chosenFile);

       looping = true;
       loopClip.loop();
                    repaint();

           }
           public void mouseExited(MouseEvent me)
```

```
        {
                repaint();
        }

        public void mousePressed(MouseEvent me)
        {
                new foree();
                mf.setVisible(false);

        }
        public void mouseReleased(MouseEvent me)
        {
        }


}

class SoundList extends java.util.Hashtable {
   JApplet applet;
   URL baseURL;

   public SoundList(URL baseURL) {
      super(5); //Initialize Hashtable with capacity of 5 entries.
      this.baseURL = baseURL;
   }

   public void startLoading(String relativeURL) {
      new SoundLoader(this, baseURL, relativeURL);
   }

   public AudioClip getClip(String relativeURL) {
      return (AudioClip)get(relativeURL);
   }

   public void putClip(AudioClip clip, String relativeURL) {
      put(relativeURL, clip);
   }
}

class SoundLoader extends Thread {
   SoundList soundList;
   URL completeURL;
   String relativeURL;

   public SoundLoader(SoundList soundList,
                URL baseURL, String relativeURL) {
      this.soundList = soundList;
      try {
```

```
            completeURL = new URL(baseURL, relativeURL);
        } catch (MalformedURLException e){
            System.err.println(e.getMessage());
        }
        this.relativeURL = relativeURL;
        setPriority(MIN_PRIORITY);
        start();
    }

    public void run() {
        AudioClip audioClip = Applet.newAudioClip(completeURL);
        soundList.putClip(audioClip, relativeURL);
    }
}
```

// Code for DATABASE QUERY

```
void query(int k)
{
          Connection con = null;
          Statement start = null;
          ResultSet rs = null;
          int i=1;
        try{

             Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
             con=DriverManager.getConnection("jdbc:odbc:puguri","scott","tiger");
            }

    catch(Exception e)
    {
                System.out.println("Problem....."+e);
    }

    try
    {
                start=con.createStatement();
                if(k==1)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug1'");
                else if(k==2)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug2'");
                else if(k==3)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug3'");
                else if(k==4)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug4'");
```

```
                else if(k==5)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug5'");
                else if(k==6)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug6'");
                else if(k==7)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug7'");
                else if(k==8)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug8'");
                else if(k==9)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug9'");
                else if(k==10)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug10'");
                else if(k==11)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug11'");
                else if(k==12)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug12'");
                else if(k==13)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug13'");
                else if(k==14)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug14'");
                else if(k==15)
                 rs=start.executeQuery("select * from spraying where
sec_code='pug15'");

                while(rs.next())
                {

                    t1.setText(rs.getString(2));
                    t2.setText(rs.getString(3));
                    t3.setText(rs.getString(4));

                }

            con.close();
    }
    catch(Exception e)
    {
            System.out.println("jprong-------"+e);
```

```
     }
    }
```

// Code to INSERT data into the Database

```
void manure_insert(int k)
  {
        Connection connec = null;
        Statement start = null;
        PreparedStatement htmt=null;
        int i=1;
      try{

                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

connec=DriverManager.getConnection("jdbc:odbc:puguri","scott","tiger");
        }

    catch(Exception e)
    {
                System.out.println("Problem....."+e);
    }

    try
    {

                htmt=connec.prepareStatement("INSERT INTO MANURING
(SEC_CODE,L_M_DATE,N_O_W,N_M_DATE) VALUES(?,?,?,?)");

                if(k==1)
                  s1="pug1";
                else if(k==2)
                  s1="pug2";
                else if(k==3)
                  s1="pug3";
                else if(k==4)
                  s1="pug4";
                else if(k==5)
                  s1="pug5";
                else if(k==6)
                  s1="pug6";
                else if(k==7)
                  s1="pug7";
                else if(k==8)
                  s1="pug8";
                else if(k==9)
                  s1="pug9";
                else if(k==10)
```

```
                s1="pug10";
               else if(k==11)
                s1="pug11";
               else if(k==12)
                s1="pug12";
               else if(k==13)
                s1="pug13";
               else if(k==14)
                s1="pug14";
               else if(k==15)
                s1="pug15";
               htmt.setString(1,s1);

               s2=t2.getText();

               htmt.setString(2,ld);
               htmt.setString(3,s2);

               htmt.setString(4,nd);

               htmt.executeUpdate();
               connec.close();
              JOptionPane.showMessageDialog((Component)null,"Data     havebeen
successfully inserted",
                              "SUBMITTED",JOptionPane.ERROR_MESSAGE);
    }

    catch(SQLException e)
    {
            JOptionPane.showMessageDialog((Component)null,"Data    is    already
inserted for this section",
                              "INSERT ERROR",JOptionPane.ERROR_MESSAGE);

            //System.out.println("jprong-------"+e);
    }
  }
```

// Code to UPDATE data of the Database

```
        void sp_update(int k)
        {
                Connection con = null;
                Statement ps = null;
                 try{

              Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         con=DriverManager.getConnection("jdbc:odbc:puguri","scott","tiger");
```

```
    ps=con.createStatement();
   }

   catch(Exception e)
   {
          System.out.println("Problem....."+e);
   }
try
{
  try
  {
  if(k==1)
           s1="pug1";
         else if(k==2)
           s1="pug2";
         else if(k==3)
           s1="pug3";
         else if(k==4)
           s1="pug4";
         else if(k==5)
           s1="pug5";
         else if(k==6)
           s1="pug6";
         else if(k==7)
           s1="pug7";
         else if(k==8)
           s1="pug8";
         else if(k==9)
           s1="pug9";
         else if(k==10)
           s1="pug10";
         else if(k==11)
           s1="pug11";
         else if(k==12)
           s1="pug12";
         else if(k==13)
           s1="pug13";
         else if(k==14)
           s1="pug14";
         else if(k==15)
           s1="pug15";
         s2=t1.getText();
         s3=t2.getText();
         s4=t3.getText();
         }
        catch(Exception e)
        {
            System.out.println("Error");
```

```
            }
            JOptionPane.showMessageDialog((Component)null,"Data is updated
successfully",
                                "UPDATION",JOptionPane.ERROR_MESSAGE);
            ps.executeQuery("update spraying set
sec_code='"+s1+"',l_s_date='"+s2+"',n_o_w='"+s3+"',n_s_date='"+s4+"' where
sec_code='"+s1+"'");
            con.commit();

        }catch(SQLException e)
        {

        }

          }
```

// Code to DELETE data from the Database

```
          void del_pluk(int k)
          {
                  Connection con = null;
                  Statement ps = null;
                  String s1=null;
                  int confirm=JOptionPane.showOptionDialog(f,"Do you really want
to Delete?","Delete Confirmation",

JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE,null,null,
null);
             if(confirm==0)
             {
               try{

                  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con=DriverManager.getConnection("jdbc:odbc:puguri","scott","tiger");
            ps=con.createStatement();

          try
           {
           if(k==1)
                    s1="pug1";
                  else if(k==2)
                    s1="pug2";
                  else if(k==3)
                    s1="pug3";
                  else if(k==4)
                    s1="pug4";
                  else if(k==5)
                    s1="pug5";
```

```
                else if(k==6)
                  s1="pug6";
                else if(k==7)
                  s1="pug7";
                else if(k==8)
                  s1="pug8";
                else if(k==9)
                  s1="pug9";
                else if(k==10)
                  s1="pug10";
                else if(k==11)
                  s1="pug11";
                else if(k==12)
                  s1="pug12";
                else if(k==13)
                  s1="pug13";
                else if(k==14)
                  s1="pug14";
                else if(k==15)
                  s1="pug15";

        JOptionPane.showMessageDialog((Component)null,"Data is deleted
successfully",
                                      "DELETION",JOptionPane.ERROR_MESSAGE);
                ps.executeQuery("delete from plucking where sec_code='"+s1+"'");
                con.commit();

        }catch(SQLException e)
        {

        }
       }catch(Exception e)
       {
       }
   }

}
```

9.4 Code Efficiency:

Efficiency is a performance requirement. When software is built and that is working up to the satisfaction of the user, the confidence of the development team goes up. While building sophisticated software the efficiency of the code, how much memory is used and in how much time it is executing, become significant. The application should optimize the code by making it smaller and speeding of calculations and displays.

Coding style can have an effect on execution speed and memory requirement. The application is build in Java Swing. The image size is a crucial factor in the speed of execution of the application. However, some conventions have been followed for making the code efficient —

- Mixing of variables have been avoided as much as possible,
- Arithmetic and logical expressions have been simplified.

10

User Instructions

10.1 Instructions to the Users of the Package

Our project is on the topic namely GIS Anchored Integrated Plantation Management: Tea, which is a user friendly GIS based package for Phuguri Tea Estate, Mirik, Darjeeling. We have tried to give best effort in making this package a user friendly, so that any one can use this package for their need without complexity.

In this project at first the user has to go through the introduction and short details about the entire work and developers of the package. Then the administrator level password checking is done for the user. If the user has administrator level access authority, then he/she can access the overall package.

After the administrator username and password checking, the main page containing the complete map of the tea garden and a button for Total Report appears in the screen. In this complete map if mouse is over on a particular section, the zoomed view of that very section can be seen. Now to work with that section, user has to left click on it. As the user click that section, the view of that section with the information corresponding to that section such as Section Code and Section Area appears from the database. Now by right clicking on the section map a pop up menu appears describing various options like **Input, Report, Search** and **Demand Analysis.** By selecting the **Input** menu the submenu items containing **Plucking, Pruning, Manuring, Spraying, Employee** and **Worker** appears. Each of the submenu have three more submenu items **Insert, Update** and **Delete** through which the user can insert, delete and also update several data for the corresponding section of the garden. Before insertion, updation or deletion of

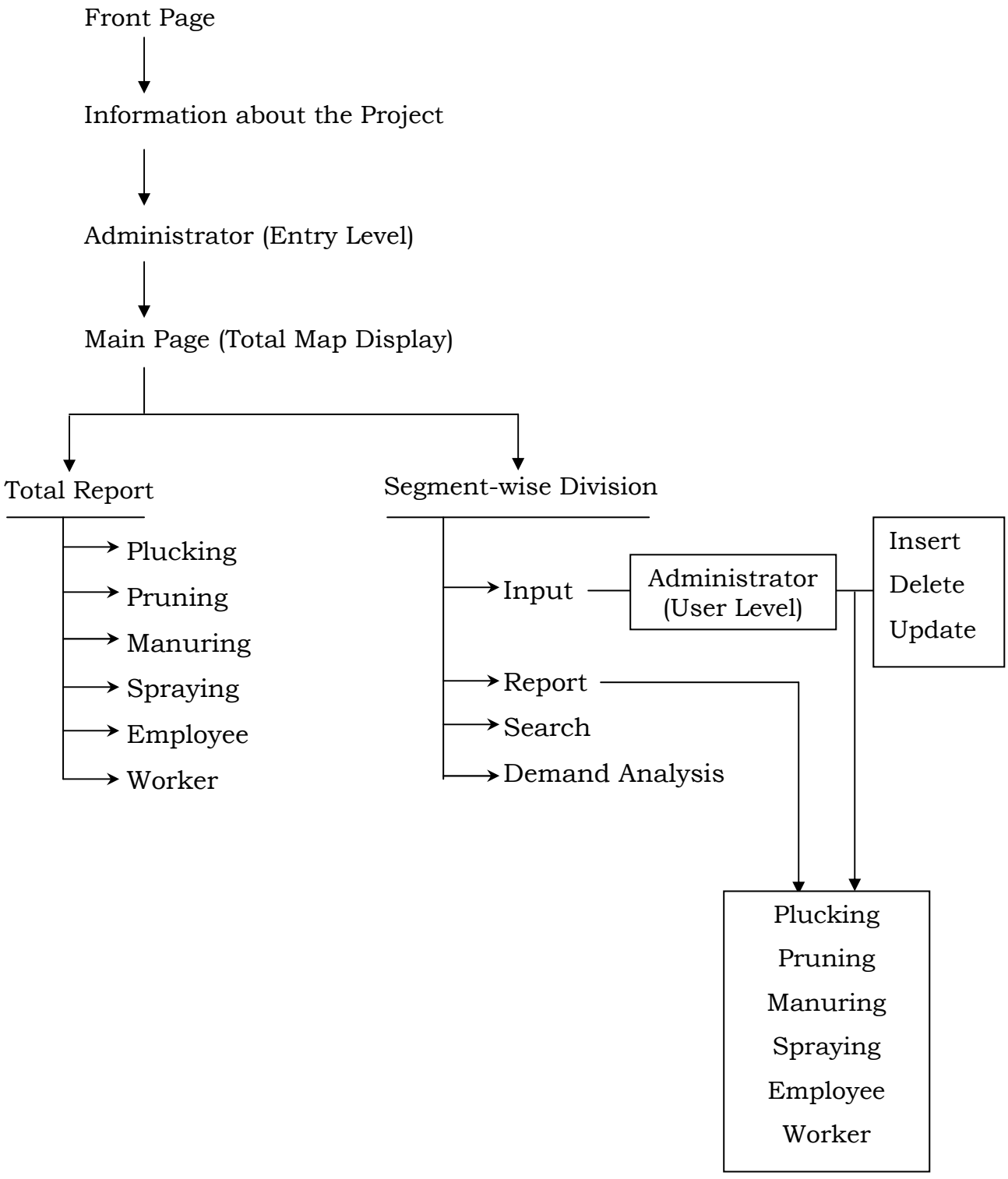
data the user has to prove his access authority by passing through his/her user name & password checking (user level).

Next by selecting the **Report** popup menu the submenu items containing **Plucking, Pruning, Manuring, Spraying, Employee** and **Worker** appear. By choosing each submenu item, the corresponding report of that very section can be viewed.

Again by selecting the **Search** popup menu different queries for any section can be asked and the information regarding that very query can be obtained.

In the main page if the **Total Report** Button is clicked then a new window appears containing various buttons like **Pruning, Plucking, Manuring, Spraying, Employee** and **Worker**. Just clicking on these buttons the user can get the detail information of the chosen field for the entire garden.

10.2 Page Settings of the Package



11

Bibliography

1. **Herbert Schildt, Java 2: The Complete Reference, Tata McGraw Hill Publishing Company Limited, New Delhi, India.**
2. **Pressman Roger S., Software Engineering — A Practitioner's Approach, McGraw Hill, Inc.**
3. **Hawryszkiewycz I. T., System Analysis and Design, Prentice-Hall of India Private Limited, India.**
4. **Robert Eckstein, Mare Loy & Dave Wood, Java Swing, O'Reilly & Associates, Inc.**
5. **User manual of Matlab 6.1**
6. **www.teaaction.com/industry/indiantea.asp**